经典面试题之 new 和 malloc 的区别

0. 属性

new/delete 是 C++关键字，需要编译器支持。malloc/free 是库函数，需要头文件支持。

1. 参数

使用 new 操作符申请内存分配时无须指定内存块的大小，编译器会根据类型信息自行计算。而 malloc 则需要显式地指出所需内存的尺寸。

2. 返回类型

new 操作符内存分配成功时，返回的是对象类型的指针，类型严格与对象匹配，无须进行类型转换，故 new 是符合类型安全性的操作符。而 malloc 内存分配成功则是返回 void * ，需要通过强制类型转换将 void*指针转换成我们需要的类型。

3. 分配失败

new 内存分配失败时，会抛出 bac_alloc 异常。malloc 分配内存失败时返回 NULL。

4. 自定义类型

new 会先调用 operator new 函数，申请足够的内存（通常底层使用 malloc 实现）。然后调用类型的构造函数，初始化成员变量，最后返回自定义类型指针。delete 先调用析构函数，然后调用 operator delete 函数释放内存（通常底层使用 free 实现）。

malloc/free 是库函数，只能动态的申请和释放内存，无法强制要求其做自定义类型对象构造和析构工作。

5. 重载

C++允许重载 new/delete 操作符，特别的，布局 new 的就不需要为对象分配内存，而是指定了一个地址作为内存起始区域，new 在这段内存上为对象调用构造函数完成初始化工作，并返回此地址。而 malloc 不允许重载。

6. 内存区域

new 操作符从自由存储区（free store）上为对象动态分配内存空间，而 malloc 函数从堆上动态分配内存。自由存储区是 C++基于 new 操作符的一个抽象概念，凡是通过 new 操作符进行内存申请，该内存即为自由存储区。而堆是操作系统中的术语，是操作系统所维护的一块特殊内存，用于程序的内存动态分配，C 语言使用 malloc 从堆上分配内存，使用 free 释放已分配的对应内存。自由存储区不等于堆，如上所述，布局 new 就可以不位于堆中。